

11/28/00  
jc961 U.S. PTO

12-01-00

Please type a plus sign (+) inside this box → PTO/SB/05 (09-00)  
Approved for use through 10/31/2002. OMB 0651-0032Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE  
PTO  
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PTO

09/27/317

11/28/00

UTILITY  
PATENT APPLICATION  
TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No. 80398P330

First Inventor or Application Identifier Hawley Knox Rising, III

Title A METHOD AND AN APPARATUS FOR IMPLEMENTING A KEY FRAME

Express Mail Label No. EL635698385US

## APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents

ADDRESS TO: Assistant Commissioner for Patents  
Box Patent Application  
Washington, DC 20231

1.  Fee Transmittal Form (e.g. PTO/SB/17)  
(Submit an original, and a duplicate for fee processing)

2.  Applicant claims small entity status.  
See 37 CFR 1.27.

3.  Specification Total Pages 20  
(preferred arrangement set forth below)  
- Descriptive title of the Invention  
- Cross References to Related Applications  
- Statement Regarding Fed sponsored R & D  
- Reference to sequence listing, a table, or a computer program listing appendix  
- Background of the Invention  
- Brief Summary of the Invention  
- Brief Description of the Drawings (if filed)  
- Detailed Description  
- Claim(s)  
- Abstract of the Disclosure

4.  Drawing(s) (35 U.S.C. 113) Total Sheets 4

5. Oath or Declaration Total Pages

a.  Newly executed (original copy)  
b.  Copy from a prior application (37 CFR 1.63(d))  
(or continuation/divisional with Box 16 completed)  
i.  **DELETION OF INVENTOR(S)**  
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).

6.  Application Data Sheet. See 37 CFR 1.76.

7.  CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)

8. Nucleotide and/or Amino Acid Sequence Submission  
(if applicable, all necessary)  
a.  Computer Readable Form (CFR)  
b.  Specification Sequence Listing on:  
i.  CD-ROM or CD-R (2 copies); or  
ii.  Paper  
c.  Statement verifying identity of above copies

## ACCOMPANYING APPLICATION PARTS

9.  Assignment Papers (cover sheet & document(s))

10.  37 CFR 3.73(b) Statement  Power of Attorney (when there is an assignee)

11.  English Translation Document (if applicable)

12.  Information Disclosure Statement (IDS)/PTO - 1449  Copies of IDS Citations

13.  Preliminary Amendment

14.  Return Receipt Postcard (MPEP 503)  
(Should be specifically itemized)

15.  Certified Copy of Priority Document(s)  
(if foreign priority is claimed)

16.  Other: .....

## 17. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:

 Continuation  Divisional  Continuation-in-part (CIP) of prior application No: \_\_\_\_\_ / \_\_\_\_\_

Prior application Information: Examiner \_\_\_\_\_

Group/Art Unit: \_\_\_\_\_

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 4b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

## 17. CORRESPONDENCE ADDRESS

 Customer Number of Bar Code Labelor  Correspondence address below

\*08791\*

Name	BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP				
Address	12400 Wilshire Boulevard, Seventh Floor				
City	Los Angeles	State	California	Zip Code	90025
Country	U.S.A.	Telephone	(310) 207-3800	Fax	(310) 820-5988

Name (Print/Type) Carol F. Barry, Reg. No. 41,600

Signature

Date

11/28/00

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

# FEE TRANSMITTAL

## for FY 2001

Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT (\$)

710.00

## Complete if Known

Application Number	80398P330
Filing Date	11/28/00
First Named Inventor	Hawley Knox Rising, III, et al.
Examiner Name	
Group Art Unit	
Attorney Docket Number	080398.P330

## METHOD OF PAYMENT (check one)

1.  The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number

Deposit Account Name

Charge Any Additional Fee Required  
Under 37CFR 1.16 and 1.17

Applicant claims small entity status.  
See 37 CFR 1.27

2.  Payment Enclosed:

Check  Money Order  Other

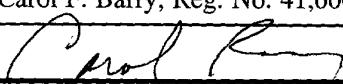
## FEE CALCULATION (continued)

## 3. ADDITIONAL FEE

Large Entity Fee Code	Small Entity Fee Code	Fee (\$)	Fee Description	Fee Paid
105	130	205	65 Surcharge - late filing fee or oath	<input type="text"/>
127	50	227	25 Surcharge - late provisional filing fee or cover sheet.	<input type="text"/>
139	130	139	130 Non-English specification	<input type="text"/>
147	2,520	147	2,520 For filing a request for ex parte reexamination	<input type="text"/>
112	920*	112	920* Requesting publication of SIR prior to Examiner action	<input type="text"/>
113	1,840*	113	1,840* Requesting publication of SIR after Examiner action	<input type="text"/>
115	110	215	55 Extension for response within first month	<input type="text"/>
116	390	216	195 Extension for response within second month	<input type="text"/>
117	890	217	445 Extension for response within third month	<input type="text"/>
118	1,390	218	695 Extension for response within fourth month	<input type="text"/>
128	1,890	228	945 Extension for response within fifth month	<input type="text"/>
119	310	219	155 Notice of Appeal	<input type="text"/>
120	310	220	155 Filing a brief in support of an appeal	<input type="text"/>
121	270	221	135 Request for oral hearing	<input type="text"/>
138	1,510	138	1,510 Petition to institute a public use proceeding	<input type="text"/>
140	110	240	55 Petition to revive - unavoidably	<input type="text"/>
141	1,240	241	620 Petition to revive - unintentionally	<input type="text"/>
142	1,240	242	620 Utility issue fee (or reissue)	<input type="text"/>
143	440	243	220 Design issue fee	<input type="text"/>
144	600	244	300 Plant issue fee	<input type="text"/>
122	130	122	130 Petitions to the Commissioner	<input type="text"/>
123	50	123	50 Petitions related to provisional applications	<input type="text"/>
126	240	126	240 Submission of Information Disclosure Stmt	<input type="text"/>
581	40	581	40 Recording each patent assignment per property (times number of properties)	<input type="text"/>
146	710	246	355 Filing a submission after final rejection (37 CFR 1.129(a))	<input type="text"/>
149	710	249	355 For each additional invention to be examined (37 CFR 1.129(b))	<input type="text"/>
179	710	279	355 Request for Continued Examination (RCE)	<input type="text"/>
169	900	169	900 Request for expedited examination of a design application	<input type="text"/>
Other fee (specify) _____				

\* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) (\$)

SUBMITTED BY					Complete (if applicable)	
Typed or Printed Name	Carol F. Barry, Reg. No. 41,600					Reg. Number
Signature		Date	11/28/00	Deposit Account User ID	02-2666	

Burden Hour Statement This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Attorney Docket No. 080398.P330  
Express Mail No. EL635698385US

UNITED STATES PATENT APPLICATION

FOR

**A METHOD AND AN APPARATUS FOR IMPLEMENTING A KEY FRAME**

Inventors:

**Hawley Knox Rising, III**  
**Ali J. Tabatabai**

Prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP  
12400 Wilshire Boulevard, Seventh Floor  
Los Angeles, California 90025  
(310) 207-3800

**A METHOD AND AN APPARATUS FOR IMPLEMENTING A KEY FRAME****BACKGROUND**

This application claims the benefit of the earlier filing date of co-pending provisional applications of Hawley K. Rising, III and Ali J. Tabatabai entitled, "A Proposed Structure for the Key Frame DS," Serial No. 60/168,433, filed November 30, 1999 and "A Method and an Apparatus for Implementing A Key Frame," Serial No. \_\_\_\_\_, filed October 20, 2000.

**FIELD OF THE INVENTION**

This invention relates generally to describing and accessing of data. More specifically, the invention relates to mechanisms and techniques that enable data related to an audiovisual work to be described and accessed.

**BACKGROUND**

Key frames have a variety of uses in terms of describing data pertaining to an audiovisual work. A key frame is a single frame that tags a plurality of frames related to a sequence of images that meet certain criteria designated by a user. For example, a description such as a title of a scene to a movie, or other information related to a particular scene may be recorded in a key frame. In this manner, the key frame is able to provide a summary of an audiovisual work that allows near random access to frames within the audiovisual work. In addition to describing audiovisual data, key frames may also be used for comparing a video with another video or for reviewing a summary of the series of frames in a document that is generated from the key frame.

One disadvantage to a key frame is that it is generally static. Once a key frame is made, the key frame generally cannot be updated. If the criteria for the key frame is changed, a new key frame must be created. Creating a new key frame is time consuming and expensive. It is therefore desirable to have a system that addresses this disadvantage.

## SUMMARY

One embodiment relates to a key frame such as a key frame description scheme (KeyFrameDS) that may be used to describe or summarize a work such as an audiovisual work based upon a criterion or criteria provided by, for example, a user. KeyFrameDS, that includes a set of attributes such as other description schemes, describes changes that are to be made to the set of description schemes. KeyFrameDS may use attribute groups containing sets of attributes such as the key, length, and value (KLV) to accomplish this task.

The KeyFrameDS may be updated in a variety of ways. In one embodiment, KeyFrameDS is updated by modifying an attribute such as the value attribute. In another embodiment, KeyFrameDS is updated by adding, deleting, or changing description schemes attached to the value attribute of a KLV attribute group. These methods for updating the KeyFrameDS allow a user to select, for example, another set of frames in an audiovisual work to provide another description or summary to the audiovisual work. Updating the KeyFrameDS may be accomplished by a sender (e.g., a server, a broadcast unit, etc.) sending a command such as a commandDS to a receiver (e.g., client, set-top box, etc.). CommandDS includes instructions such as to add, change, or delete one or more attributes or to add, change or delete a description scheme.

One example of updating a KeyFrameDS relates to a person, driving in a vehicle with a portable computer device, who is initially interested in touring historical sites in a city. A KeyFrameDS, that includes the attribute groups  $K_1L_1V_1$ , may be used to provide a summary of these historical sites. At about noon, the person may be interested in finding restaurants in the city. In order to view a list of restaurants, the person inputs information that changes the value attribute ( $V_1$ ) in the  $K_1L_1V_1$  attribute group in the KeyFrameDS that searches historical sites to restaurants. The length of bytes associated with the value attribute typically changes when the value attribute changes, so the attribute  $L_1$  is modified to attribute  $L_2$ . The remainder of the information in the KLV attribute group of the KeyFrameDS remains unchanged by the user

such as the city in which the search is to be performed. The computer system executes this request and presents a list of restaurants to the user.

Another embodiment of KeyFrameDS involves placing entities such as other description schemes into the value attribute of a KLV attribute group in a KeyFrameDS in a universally recognizable format, such as in a description definition language (DDL). By using KeyFrameDS that have these characteristics, the value attribute of the KeyFrameDS may be modified regardless of syntactic or semantic distinctions that may exist between, for example, semantic data that describes a syntactic audiovisual object. Other features and advantages of the invention will be apparent from the accompanying drawings and from the detailed description that follows below.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements, and in which:

**Figure 1** illustrates a block diagram of one embodiment of a computer system that includes servers;

**Figure 2** illustrates a block diagram of one embodiment of another computer system related to a peer-to-peer system;

**Figure 3** illustrates a block diagram of another embodiment of a computer system; and

**Figure 4** illustrates a flow diagram of one embodiment for updating a key frame.

#### **DETAILED DESCRIPTION**

When creating an audiovisual work, it is desirable to provide a description or a summary of portions of the audiovisual work that contain certain criteria designated by a person such as a user of a computer system. The criteria may be changed to incorporate different information. To accommodate the changed criterion or criteria, one embodiment allows a key frame such as a key frame description scheme (KeyFrameDS) to be used.

KeyFrameDS is a description scheme that includes other description schemes. To adjust for the changed criteria, the KeyFrameDS is updated.

To understand the manner in which the KeyFrameDS may be modified, an explanation of the structure of the KeyFrameDS is provided.

The KeyFrameDS includes a list of attribute groups. Each attribute group includes a key, length, and value attribute (KLV). The key attribute, typically an object/class, describes other description schemes. For instance, the key attribute may indicate that the key frame relates to the rules of a soccer game that is to be broadcast. The length attribute generally refers to the number of bytes associated with the value attribute. The value attribute, that includes one or more elements, is used to instantiate the object or class provided by the key attribute. The value attribute is where a set of description schemes attach thereto. Typically, the value attribute and the length attribute are modified whereas the key attribute generally remains unchanged when values are altered. If the description schemes themselves are altered, the key attribute changes to reflect the new description scheme name.

Updating the KeyFrameDS includes adding, deleting, or modifying attributes in attribute groups contained in KeyFrameDS. There are at least two general ways in which the KeyFrameDS is updated. First, the value contained in a value attribute of a KLV attribute group of the KeyFrameDS may be modified. Second, the value attribute may be modified by updating the description schemes contained in or pointed to by the value attribute. Updating the KeyFrameDS may be accomplished by a sender (e.g., a server, a broadcast unit, etc.) sending a command such as a commandDS to a receiver (e.g., client, set-top box, etc.).

The commandDS contains information for making a change to at least one attribute of the KeyFrameDS or add, delete, or change a description scheme. The commandDS may require that the change occur at read time or any other suitable time.

By implementing techniques of the invention, a user may update a KeyFrameDS during, for example, a broadcast of a sports program on a

television. Additionally, the state of the descriptions may change to reflect a change to the underlying audiovisual content.

In the following description, numerous specific details such as specific materials, processing parameters, processing steps, *etc.*, are set forth in order to provide a thorough understanding of the invention. One skilled in the art will recognize that these details need not be specifically adhered to in order to practice the claimed invention. In other instances, well known processing steps, materials, *etc.*, are not set forth in order not to obscure the invention.

**Figures 1 through 3** illustrate computer systems that may implement various features of a KeyFrameDS to change attributes such as description schemes. To better understand techniques of the invention that are implemented using various computer systems, definitions of terms are provided. A description scheme is a characteristic associated with the data and may be the value of a value attribute. Such attributes may include syntactic description schemes (*e.g.*, arrangement of words), semantic description schemes (*e.g.*, meanings of words), a mixture of these description schemes, or any other suitable characteristics of data. The attributes of the KeyFrameDS may be dynamically changed by a user, a server, a client, or any other suitable device.

The KeyFrameDS includes attributes such as key, length, and value (KLV), grouped into the KLV attribute groups. Key attribute is a tag that uniquely identifies an object/class (*e.g.*, a set of data with similar characteristics). A key attribute describes a description scheme and typically remains unchanged. Length attribute is the number of bytes associated with the value attribute. The length attribute may be skipped if a receiver (*e.g.*, client, set-top box, etc.) of the KeyFrameDS does not recognize the object/class thereby reducing the amount of time to process a KeyFrameDS. The value attribute, on the other hand, is used to instantiate the object/class.

Given this description of KeyFrameDS and attributes, a block diagram of one embodiment of computer system 1 that implements principles of the invention is presented in **Figure 1**.

## I. SERVER IMPLEMENTATION OF TECHNIQUES OF THE INVENTION

KeyFrameDS interactions may be implemented on a server-side through servers (2, 4) and on the client-side through clients (6, 8). Each of these implementations is discussed below.

### A. SERVER-SIDE INTERACTION WITH A KEY FRAME

Computer system 1 includes servers (2, 4) and clients (6, 8). Clients (6, 8) are applications that operate on a personal computer or a workstation to input data used in key frames. Servers (2, 4) are computers or devices on a network that manage network resources. For example, server 2 is a computer and storage device dedicated to the task of storing data and searching KeyFrameDS. Program instructions (e.g., computer program, software, etc.) reside in server 2. Program instructions that may enhance techniques of the invention include a standard generalized markup language such as extensible markup language schemes (XML) and description definition language (DDL). Scheme XML uses two types of data items such as elements and attributes in order to change a description in, for example, a KeyFrameDS, related to an audiovisual work.

Another embodiment of KeyFrameDS involves placing the key frame in a universally recognizable format, such as in the DDL. By using KeyFrameDS that have these characteristics, an attribute group such as the value attribute group of the KeyFrameDS may be modified regardless of syntactic or semantic distinctions that may exist between, for example, semantic data that describes a syntactic audiovisual object. In comparison to the functions of server 2, server 4 may be a file server, print server, or any other suitable server to perform certain functions.

On the server-side, a mechanism is used for updating KeyFrameDS. There are at least two general ways in which a KeyFrameDS is updated. In one embodiment, KeyFrameDS is updated by modifying an attribute such as the value attribute. In another embodiment, KeyFrameDS is updated by attaching, adding, deleting, or changing description schemes.

Updating the KeyFrameDS may be accomplished by a sender such as server 4 sending a commandDS to a receiver such as server 2. CommandDS contains the information to update the KeyFrameDS. For example, commandDS includes changes to at least one description scheme by indicating that a KLV attribute group, for instance,  $K_1L_1V_1$ , should be changed to the KLV attribute group  $K_1L_2V_2$ . CommandDS identifies that the key is  $K_1$  which is the same as the key attribute in the original KLV attribute group in the original KeyFrameDS. This identifies the particular key frame (e.g., KeyFrameDS) to be updated and, in particular, the attribute to be updated in the KeyFrameDS.  $L_2$  is the length of the new value attribute  $V_2$ .

CommandDS allows a receiver to add, change, or delete parts of a description (e.g., an audio description scheme related to descriptions of sound) dynamically based upon the user preferences (e.g., recordation of a title of a movie, important scenes in a movie, etc.) and or domain-specific attributes. The commandDS may also contain information for making the change during read time or at any other suitable time. The code for commandDS is as follows:

```

<datatype name='commandTypes'>
    <basetype name='string' />
    <enumeration>
        <literal>add</literal>
        <literal>change</literal>
        <literal>delete</literal>
    </enumeration>
</datatype>
<DSType name = 'CommandDS'>
    <attribute name = 'command' type ='commandTypes'>
        minOccurs='1' maxOccurs='1' />
        <attrGroupRef name = 'KLV' />
    </DSType>

```

As noted in the commandDS code, the instructions include, for example, to add, change, or delete information associated with KeyFrameDS.

CommandDS, executed on server 2, may automatically update the KeyFrameDS and then search for data in a database such as that which is stored on server 2 that complies with the requirements established by the new KeyFrameDS. To illustrate, commandDS may require that an attribute for the KeyFrameDS be modified based upon a factor such as time (e.g., every fifteen seconds an attribute such as the value attribute automatically changes based upon program instructions executed on the server).

The commandDS may require that an attribute or attributes (e.g., description schemes) attached to the KLV attribute group of a key frame such as KeyFrameDS change, for example, every fifteen seconds. To illustrate, a KeyFrameDS such as gameKeyFrameDS may be used in a sports program such as a soccer game. If a soccer game is being played, one screen of a split screen may be used to display information pertaining to the soccer players.

GameKeyFrameDS may be used to automatically control the type of information displayed on, for example, a graphical user interface (GUI) of client 6. GameKeyFrameDS may include a plurality of attributes such as a player description scheme (playerDS), soccer description scheme (soccerDS), audio description scheme (audioDS), and transcoding description scheme (transcoding DS). Each description scheme may appear on a GUI as a hyperlink. In order to automatically change information related to a soccer player, an attribute such as the playerDS may be used that links the player's name to information such as the player's age, height, weight, and a short summary of the player's career history. Other information that may be related to playerDS includes a color descriptor to describe the color of the player's uniform.

Server 2 could automatically change the attribute such as playerDS every fifteen seconds by changing the name of a first player to the name of a second player player. The player's name may be changed by, for instance, the program instructions alphabetically rotating players last name every fifteen seconds. Information such as age, height, weight, and career history that is associated with the name of the player is automatically updated when the player's name is changed. This new information is then sent over

interconnect 13 to client 6. A user may then view the new gameKeyFrameDS and the data associated with the gameKeyFrameDS.

Other attributes that may be automatically modified by the program instructions include other descriptive schemes such as soccer description scheme (soccerDS), audio description scheme (audioDS), transcoding description scheme (transcodingDS), and time description scheme (timeDS). Each of these descriptive schemes is described below.

SoccerDS contains descriptions related to the rules associated with the game of soccer. For example, this attribute may indicate that the soccer ball must be hit by a player into an area defined by goal posts and a net in order for a point to be added to the team's score. Another rule may be that a soccer player cannot hold the soccer ball in his or her hands. Various other suitable rules may be incorporated into this attribute.

AudioDS contains audio related descriptions for sound such as sound effects, instruments, speech recognition, music, or any other suitable audio description. For example, audioDS may provide the voice of the sportscaster broadcasting the event.

TranscodingDS contains descriptions related to the coding type of the audiovisual work such as a picture. For example, transcoding may involve converting from a picture format to another format (*e.g.*, Moving Picture Experts Group - 2 Standard (MPEG-2) to H. 263).

TimeDS is generally composed of two elements: the start time point and the duration of a particular segment. Time stamping is used to mark areas in a multimedia work. To illustrate, a time stamp may be used to mark a certain play in the soccer game in order to allow that play to be replayed for the audience. Any one of these description schemes may be automatically modified by server 2 as explained herein.

In addition to updating attributes (*e.g.*, description schemes), another embodiment relates to weighting attributes of a KeyFrameDS. For example, one attribute may be assigned a weight of "1" that means very important such as playerDS and timeDS while the weight of "5" is assigned to another attribute and means least important such as soccerDS.

In another example, an attribute such as a group name may be considered very important. Therefore, this attribute is assigned a value of importance of "1" such as for the key and length code provided below.

```

<attrGroup name = 'REF_ID'
5           <attribute name = 'id' type = 'ID' minOccurs='1' maxOccurs='1' />
           <attribute name = 'href' type = 'uri' minOccurs='1'
           maxOccurs='1' />
         </attrGroup>
         <attrGroup name = 'KLV'
10          <attribute name = 'key' type = 'ID' minOccurs='1'
           maxOccurs='1' />
           <attribute name = 'length' type='integer' minOccurs='1'
           maxOccurs='1' />
           <attribute name = 'value' type ='char' maxOccursPar='length' />
         </attrGroup>
         <datatype name='attributeWeight'>
           <basetype name='integer' />
           <minInclusive> 1 </minInclusive>
           <maxInclusive> 5 </maxInclusive>
         </datatype>
         <DSType name = 'KeyFrameDS'>
           <attrGroupRef name = 'REF_ID' />
           <SubDSOf name = 'keyFrame' />
           <attribute name = 'size' type = 'int' />
25           <seq minOccurs = '1' maxOccursPar = 'size' />
           <attrGroupRef name = 'KLV' />
           <attribute name = 'weight' type = 'attributeWeight' minOccurs =
           '0' maxOccurs = '1' />
           </seq>
         </DSType>
30

```

15  
20

25

30

By allowing attributes to be weighted, server 2 or a user of a computer system is able to dynamically determine, for example, the types of attributes that should be more frequently displayed to a user.

5 Additionally, KeyFrameDS may mark an audiovisual work through a reference identifier (ID) by using methods known in the art. A reference ID indicates a certain location in an audiovisual work. A reference ID may include a media or a medium locator to specify the "location" of a particular image, audio, or video segment by referencing the media data. There are generally four types of medium locators such as the video segment locator, 10 the audio segment locator, the image locator, and the sound locator. In this manner, a user may randomly access frames designated with a reference ID. It will be appreciated that other methods may be used to mark a multimedia work such as an audiovisual work.

15

#### **B. CLIENT-SIDE INTERACTION WITH DYNAMIC KEY FRAME**

Another implementation of updating the KeyFrameDS may be performed by a user inputting and sending changes using a commandDS through, for example, client 6 to a server of the computer system shown in Figure 1. Clients (6, 8) rely upon a server such as servers (2, 4) to perform certain operations such as to input a commandDS to update KeyFrameDS in order to access previously stored information related to the KeyFrameDS. Server 2 also allows or causes information to a KeyFrameDS to be added, changed, or deleted.

25 The KeyFrameDS may be updated by having one of its attribute groups or weights changed such as the KLV (key, length, and value) attribute groups, or their corresponding weights. Generally, a user will modify a KLV attribute group by modifying the value attribute since the value attribute incorporates other attributes such as description schemes. For example, information such as the shot identification, the scene identification, and key-frame building, or 30 selecting parameters may be modified by the user through client 6. If a shot/scene identification is to be changed, server 2 resegments the video, selects new key frames at the desired segment quantization level, and sends

these KeyFrameDS to client 6. The user may input attributes such as description schemes that are semantic, syntactic, or a mixture of those attributes into a client such as clients (6, 8). One means for inputting this information is by a user sending from client 6 a command such as commandDS to server 2.

5 Server 2 receives and executes these instructions such as commandDS from client 6. Server 2 then may access information that has been stored in server 2 or server 4 such as data that is responsive to the KeyFrameDS. For instance, a very large collection of frames that may include a variety of possible parameter settings may be stored and accessed on server 2. Server 2 selects segments from the stored KeyFrameDS that represent those segments and that fit the parameters inputted by a user. Server 2 then sends this new KeyFrameDS and data corresponding to the new KeyFrameDS to client 6.

10 On the other hand, if the segments are to remain unchanged, the process reselects the KeyFrameDS for each segment and sends these KeyFrameDS' to client 6. If a query is processed, the KeyFrameDS are not in temporal order and the reordering of the KeyFrameDS is assumed.

15 The KeyFrameDS structure, that uses DDL, provides greater flexibility by allowing attributes such as semantic, syntactic, a mixture of semantic and syntactic attributes, or any other suitable characteristics of data to be added, changed, or deleted by automatically modifying the attribute groups in the KeyFrameDS or by a user sending a commandDS that instructs that certain changes be made. It will be appreciated that the KeyFrameDS structure and the use of commandDS to change a content of the key frame structure may 20 also be extended to define other description schemes with similar requirements.

## 25 **II. PEER-TO-PEER SYSTEM IMPLEMENTATION OF TECHNIQUES OF THE INVENTION**

30 It will be appreciated that a peer-to-peer system illustrated in **Figure 2** may be used in the KeyFrameDS interactions. Peer-to-peer system 3 is a type of network in which each workstation such as servers (14, 15) have equivalent capabilities and responsibilities. For example, there may be two peer

computers or servers on the same network. This differs from client/server architectures, in which some computers are dedicated to serving the others. Peer-to-peer networks are generally simpler and less expensive, but they usually do not offer the same performance under heavy loads.

5        In one embodiment, servers (14, 15) are coupled through interconnect 13. Client 6 may access the KeyFrameDS to add, delete, or change a value for an attribute or an element for an attribute in KeyFrameDS. A commandDS may be sent to client 6 from server 14 to update KeyFrameDS. Once client 6 has the initial list, updates such as in the form of adding, deleting, or 10 modifying a KeyFrameDS may be sent from client 6 to server 14. Server 14 then updates or changes the description scheme(s) as described above. Server 15 may also perform in the same manner.

### 15        **III. COMPUTER SYSTEM IMPLEMENTATION OF TECHNIQUES OF THE INVENTION**

20        **Figure 3** illustrates an embodiment of another computer system 100 that implements the principles of the invention. Computer system 100 includes a stand alone or portable computing device. Computer system 100 comprises a processor 170, a storage device 180, and interconnect 150 such as a bus or a point-to-point link. Processor 170 is coupled to the storage device 180 by interconnect 150. In addition, a number of user input/output devices, such as a keyboard 120 and display 125, are coupled to chip set (not shown) which is then connected to processor 170. The chipset (not shown) is typically 25 connected to processor 170 using an interconnect that is different from interconnect 150.

30        Processor 170 represents a central processing unit of any type of architecture (e.g., the Intel architecture, Hewlett Packard architecture, Sun Microsystems architecture, IBM architecture, etc.), or hybrid architecture. In addition, processor 170 could be implemented on one or more chips. Storage device 180 represents one or more mechanisms for storing data such as the plurality of elements that make up an attribute which may be incorporated into a key frame such as KeyFrameDS. Storage device 180 may include read only memory (ROM), random access memory (RAM), magnetic disk storage

media, optical storage media, flash memory devices, and/or other machine-readable media. Interconnect 150 represents one or more buses (e.g., accelerated graphics port bus, peripheral component interconnect bus, industry standard architecture bus, X-Bus, video electronics standards association related to buses, etc.) and bridges (also termed as bus controllers).

5 While this embodiment is described in relation to a single processor computer system, the invention could be implemented in a multi-processor computer system. In addition to other devices, one or more of a network 130 may be present. Network 130 represents one or more network connections for transmitting data over a machine readable media. The invention could 10 also be implemented on multiple computers connected via such a network.

15 **Figure 3** also illustrates that the storage device 180 has stored therein data 135 and program instructions (e.g., software, computer program, etc.) 136. Data 135 represents data stored in one or more of the formats described herein. Program instructions 136 represents the necessary code for performing any and/or all of the techniques described with reference to **Figures 1, 2, and 4**. It will be recognized by one of ordinary skill in the art that the storage device 180 preferably contains additional software (not shown), which is not necessary to understanding the invention.

20 **Figure 3** additionally illustrates that the processor 170 includes decoder 140. Decoder 140 is used for decoding instructions received by processor 170 into control signals and/or microcode entry points. In response to these control signals and/or microcode entry points, decoder 140 performs the appropriate operations.

25 **Figure 4** illustrates a flow diagram of one embodiment for updating a key frame. At block 200, at least one attribute group, a set of attribute groups, or information pertaining to at least one description scheme is provided for an attribute such as KeyFrameDS. Attributes such as description schemes are attached to the value attribute to the KeyFrameDS. Attributes may be in the form of semantic, syntactic, or any other suitable characteristic of data.

30 At block 210, attribute groups are inserted into a key frame such as the KeyFrameDS. At block 220, attributes such as other description schemes are

attached to an attribute group such as the value attribute . In one embodiment, a universally recognizable format such as DDL is used to insert the attributes into a KeyFrameDS. At block 230, the KeyFrameDS may be modified, for example, by a server, a user, a client, or other suitable device. A 5 command such as a commandDS is sent to, for instance, a server that requires the server to process the KeyFrameDS to change information about the KeyFrameDS. At block 240, at least one attribute group of the KeyFrameDS may be updated. Typically, when updating a KLV attribute group, the value attribute or the length attribute are modified whereas the key attribute that describes the other description schemes generally remains unchanged, unless one is changing the description scheme to which the KLV attribute group refers. Alternatively, at least one attribute is added, deleted, or changed that is 10 attached to the KeyFrameDS. At block 250, a key frame such as KeyFrameDS is processed. At block 260, the server or processor accesses stored information pertaining to the KeyFrameDS. This information may be stored in a storage medium or media such as a database or any other suitable means. In another embodiment, the server may connect to a network such as the Internet to access information.

20 In the foregoing specification, the invention is described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

**CLAIMS:**

What is claimed is:

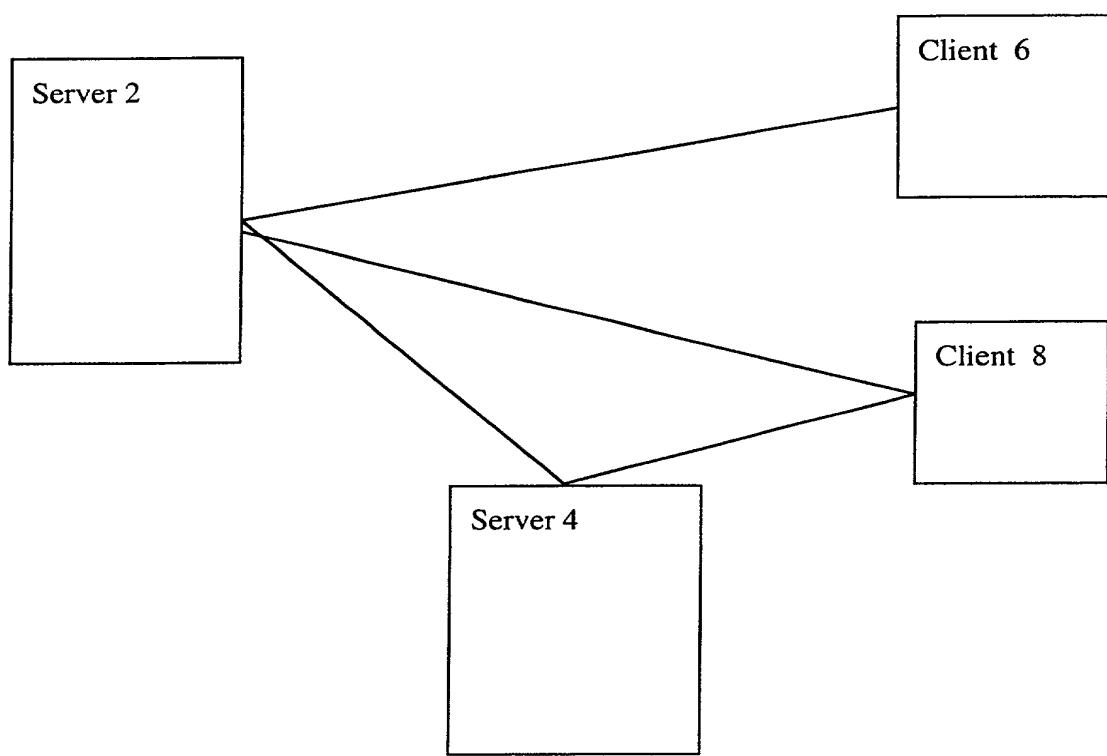
1. 1. A method comprising:
  2. providing a key frame description comprising a list of KLV attribute groups, each KLV attribute group comprising a key attribute, a length attribute, and a value attribute; and
  3. updating a key frame by performing at least one of:
    4. modifying a KLV attribute group by modifying one of a key attribute, a length attribute, and a value attribute, and modifying a weight attached to at least one KLV attribute group.
5. 2. The method of claim 1, further comprising:
  6. placing a set of attributes into a universally recognizable format.
7. 3. The method of claim 1, further comprising:
  8. sending a command to change one of the key attribute, length attribute, and value attribute of the key frame.
9. 4. The method of claim 1, further comprising:
  10. assigning a weight value to at least one attribute.
11. 5. The method of claim 3, further comprising:
  12. sending a command by one of a user, a client, and a server.
13. 6. An article comprising:
  14. a storage medium including instructions stored thereon which when executed causes a computer system to perform a method comprising:
    15. providing a key frame comprising a list of KLV attribute groups, each KLV attribute group comprising a key attribute, a length attribute, and a value attribute;
    16. updating a key frame by performing at least one of:
      17. modifying a KLV attribute group by modifying one of a key attribute, a length attribute, and a value attribute, and modifying the weight attached to at least one KLV attribute group.

- 1       7. The article of claim 6, wherein the method further comprises:  
2           configuring each attribute to include a plurality of attribute groups.
  
- 1       8. The article of claim 7, wherein the method further comprises:  
2           placing a set of attributes in a universally recognizable format.
  
- 1       9. The article of claim 7, wherein the method further comprises:  
2           sending a command to change information in the key frame.
  
- 1       10. The article of claim 6, the method further comprises:  
2           updating at least one attribute.
  
- 1       11. The article of claim 6, wherein the method further comprises:  
2           assigning a weighted percentage to at least one attribute.
  
- 1       12. The article of claim 9, wherein the method further comprises:  
2           updating at least one attribute which is one of a syntax attribute and a  
3           semantic attribute.
  
- 1       13. A computer system comprising:  
2           a processor coupled to a memory, the memory having stored therein  
3           instructions which when executed by the processor cause the processor to  
4           generate data and to:  
5               attaching a set of attributes in a key frame; and  
6               modifying the key frame by one of a user, a client, and a server;  
7           an interconnect coupled to the processor and the memory to allow the  
8           data to be transported between the memory and the processor.
  
- 1       14. The system of claim 13, wherein the set of attributes are placed in a  
2           universally recognizable format.
  
- 1       15. The system of claim 13, further comprising:  
2           a command is sent from an input/output device to the processor to  
3           change information in the key frame.

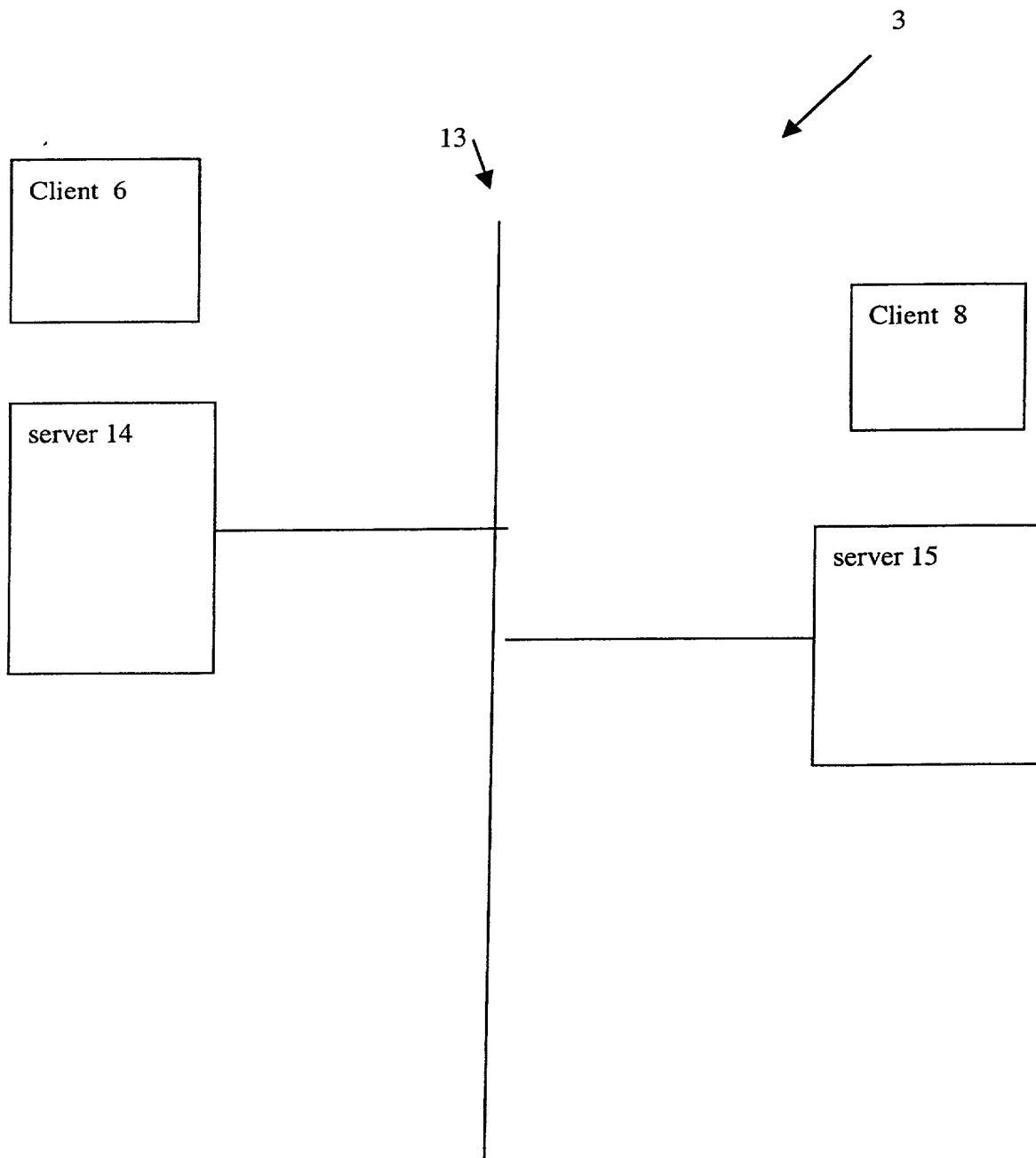
- 1        16.    The system of claim 13, wherein at least an attribute is assigned a
- 2        weight value.

**ABSTRACT OF THE DISCLOSURE**

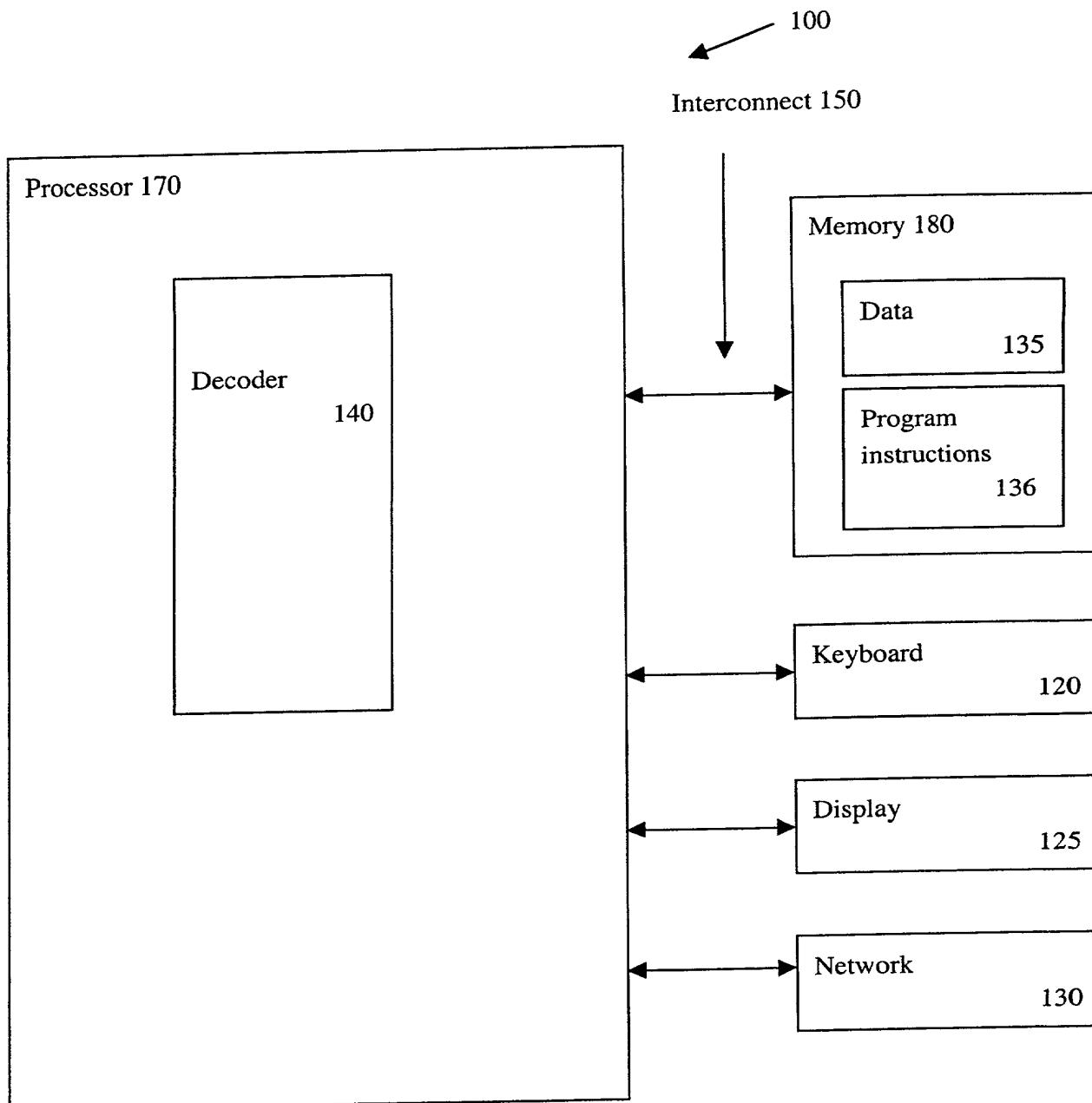
The invention relates to a key frame such as a KeyFrameDS. A list of key, length, and value (KLV) attribute groups are used in the KeyFrameDS. The KeyFrameDS may be modified by changing, for example, an attribute group. This may be accomplished, for example, by changing one of its attributes, such as the value attribute.



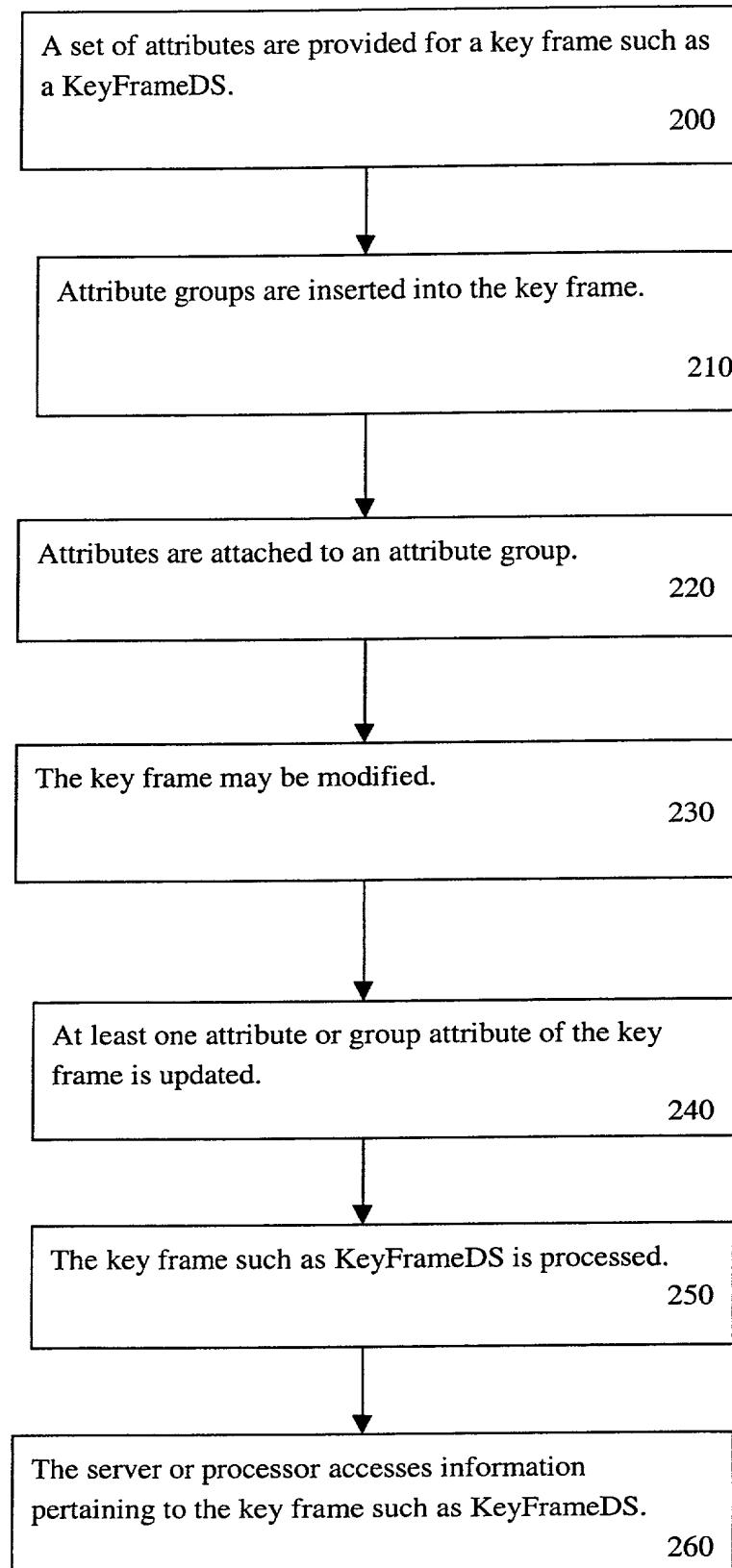
**Fig. 1**



**Fig. 2**



**Fig. 3**



**FIG. 4**